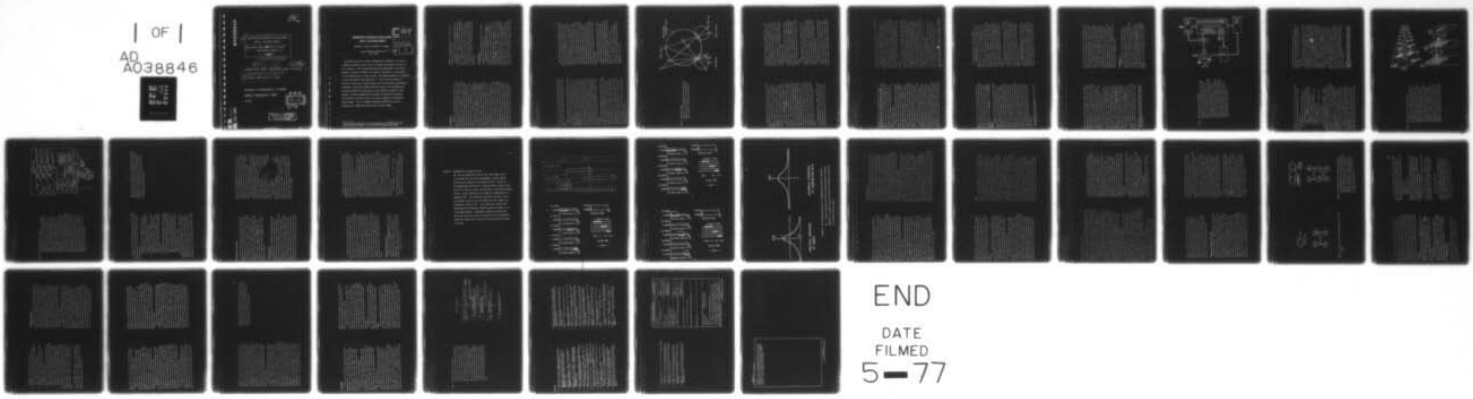
AD-A038 846 MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INF--ETC F/G 9/2
COMPUTATIONAL TECHNIQUES IN VISUAL SYSTEMS. PART I. THE OVERALL--ETC(U)
JUL 76 M A ARBIB, E M RISEMAN N00014-75-C-0459

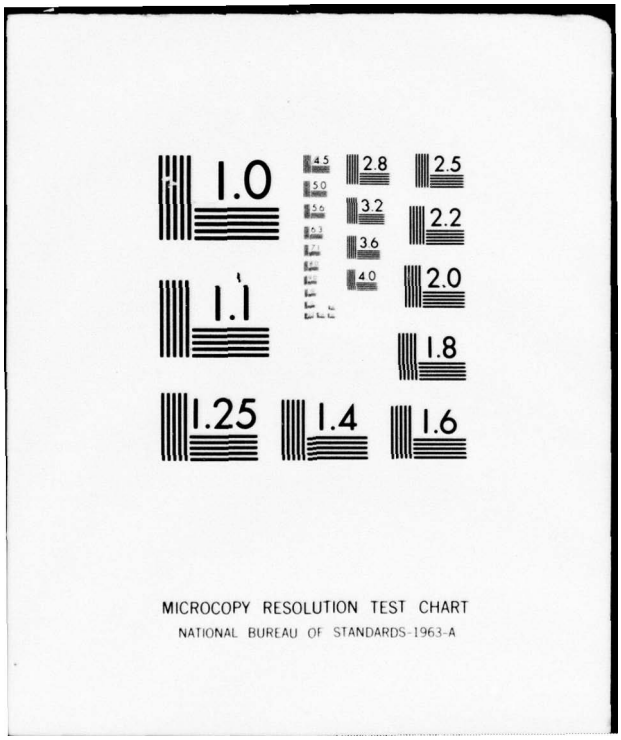
UNCLASSIFIED COINS-TR-76-10 NL

| OF |
AD
A038846



END

DATE
FILMED
5-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12 NW

# COMPUTATIONAL TECHNIQUES IN VISUAL SYSTEMS, PART I. THE OVERALL DESIGN.

Michael A. Arbib and Edward M. Riseman

COINS TECHNICAL REPORT 76-10

July 76

14 COINS-TR-76-10

9 Technical Rept.,

12 34p.

COMPUTER AND INFORMATION SCIENCE

15 N00014-75-C-0459,
PHS-NS-09755-06

UNIVERSITY OF MASSACHUSETTS AT AMHERST

AMHERST, MASSACHUSETTS 01002

U.S.A.

# COMPUTATIONAL TECHNIQUES IN VISUAL SYSTEMS

## PART I. THE OVERALL DESIGN[1]

Michael A. Arbib and Edward M. Riseman

COINS TECHNICAL REPORT 76-10
July 1976

Our overall goal is to define computational techniques to be used by a system in making a visual scan of a dynamic environment with which it is to interact. Here, we discuss both brain mechanisms in the visual systems of animals and humans and computer techniques for the analysis of color photographs of natural scenes. We present schemas as a formalization of the system's 'knowledge units'. This notion is helpful for our work in both the BT (Brain Theory) and AI (Artificial Intelligence) approaches. We further present specific studies--from our own group and from elsewhere--of subsystems of both animal and computer visual systems. We shall examine the interaction of high-level processes with low-level systems, as part of a general emphasis on integrated system design. Part II (Riseman and Arbib [1976]) will focus on techniques for segmenting single static colored images.

[1] This work was supported in part by NIH grant 5R01 NS09755-06 COM and by ONR grant N00014-75-C-0459 and NSF grant DCR75-16098.

## 1. Introduction

A human drives a car without accident down a busy street; a frog snaps up a fly; a computer system locates different resources by scanning satellite photographs; and a robot uses input from a TV camera to guide its manipulation of objects on an assembly line. In each case, a pattern of visual input (possibly a simple black-and-white photograph; possibly enriched by color and depth information; possibly changing over time) must be analyzed to yield an internal representation of the world. The visual input is itself a 'low-level' representation in terms of the brightness of light at each point and time. The internal representation is a 'high-level model' which is semantic in the sense that meaning has been ascribed to regions of the visual scene, which is now interpreted as a collection of objects in space. Each object may be represented by a name, by a more detailed description, or by a program for the system's interaction with the object. The internal model will also contain information about the state of, and the relationship between, the objects.

Before going on, we should note how much the internal model will depend on not only the nature but also the goals of the system. Clearly, a frog 'sees' a very different world from that of the assembly-line robot. More subtly, what we see depends to a great extent on what we are looking for--in recognizing a house, we may rarely perceive the type and location of the windows unless intent on breaking and entering, or on washing the windows. Again, while the representation of visual input will enable an animal or robot to answer questions about a given (possibly changing) scene, and aid in the generation of plans for interacting with and



manipulating the environment, partial plans may well determine the directions of the system's attention.

Our emphasis in this paper is on image understanding--the process of visual perception in animals, and of scene analysis in computers and robots. There is an overlapping field of research called image processing which, for example, provides ways to enhance contrast and remove noise to make an image easier for a human to use. We shall only study such processing to the extent that it provides the front end of a system--be it a neural network or computer--which comes to name significant regions in the image, or to plan patterns of interaction with the world that the image represents.

### Cooperating Systems

In analyzing image understanding systems, it helps to break the computations into low-level and high-level tasks.

Low-Level Systems perform feature extraction and segmentation. The raw representation of the environment in terms of measurements of visual input is replaced by a representation in terms of local features based on boundary, depth, motion, color, and texture cues. Further processes then segment the scene (aggregate the features) into relatively large regions each delimited by a continuous boundary, or by a consistent pattern of depth, motion, and/or texture. An image understanding system cannot function unless the wide range of colors and intensities in a tree can be viewed as a unit. Neither people's clothes, the surface of streets, a wispy cloudy sky, nor buildings, fields, or water have uniform visual characteristics of hue, saturation and intensity. Natural variations

in surface color are compounded by reflections and shadows due to the irregularity of the surface, position and type of light source, and the effect of nearby objects. (Beck [1975] provides a useful discussion of the complexity of these effects.)

We shall see segmentation techniques that form boundaries via the gradients of edges as well as techniques that grow regions on the basis of cues of color, texture, depth or motion. In each case, the segmentation processes do not operate directly on the 'raw' visual input, but rather operate on an array of preprocessed features. Two classic papers in neurophysiology show that the peripheral visual systems in animals preprocess the input, rather than simply transmitting the 'raw' input back to the brain. Lettvin, Maturana, McCulloch and Pitts [1959] showed that the frog's retina sends back to the brain several arrays of coded features--with one feature being the presence of a 'bug-like' small wiggling object. Note well that this is a motion feature, and cannot be extracted from a single static scene. Hubel and Wiesel [1962] showed that the cat retina enhanced contrast (excellent preprocessing for boundary detection) while visual cortex had cells which responded best to line stimuli of a given orientation.

Further studies showed that depth is also a significant feature for the activation of such cells. While an animal has many high-level cues for recognition of depth--such as parallax or the apparent size of objects--and a computer attached to a camera may use a range-finder, the dominant depth cue at the level of preprocessed features in the visual systems of animals with forward looking eyes is retinal disparity. To see how this is achieved, note that a single photograph (or the input to a single eye) maps the spatial direction of image points, but does not represent the distance to the stimulus point. However, two photographs of the same static scene can, if taken from different angles, provide depth information. We can see this by analyzing the visual system of an animal with stereopsis.



While each retina provides only a two-dimensional map of the visual world, the two retinae between them provide information from which can be reconstructed the three-dimensional location of all non-occluded points in visual space. We indicate this in Figure 1 where the right retina can not distinguish A, B or C ($A_R = B_R = C_R$), and where the left retina can distinguish them but cannot determine where they lie along their ray. The two retinae can actually locate them on the ray: $(A_L, A_R)$ fixes A, $(B_L, B_R)$ fixes B, and $(C_L, C_R)$ fixes C. In fact, '*depth detectors*' which combine information from the two input patterns to determine the three-dimensional location of a point have been found in the brain of cats and monkeys. Barlow, Blakemore and Pettigrew [1967], Pettigrew, Nikara and Bishop [1968], and others found cells in visual cortex which not only respond best to a given orientation of a line stimulus (as shown by Hubel and Wiesel) but do so with a response which is sharply tuned to the disparity of the effect of the stimulus upon the two retinae.

We have already seen that animals have motion detectors. Computer systems, too, will benefit from the extra cues that motion provides. As we know from our experience with television and motion pictures, continually changing visual input can be approximated without perceptual deficit by a sufficiently rapid sequence of frames (in the usual cinematographic sense). With this dynamic input, the system should not be overburdened by a complete set of static features for every frame. Rather, the system should use features generated by motion detectors--a set of co-moving points being prime candidates for segmentation into a single region.

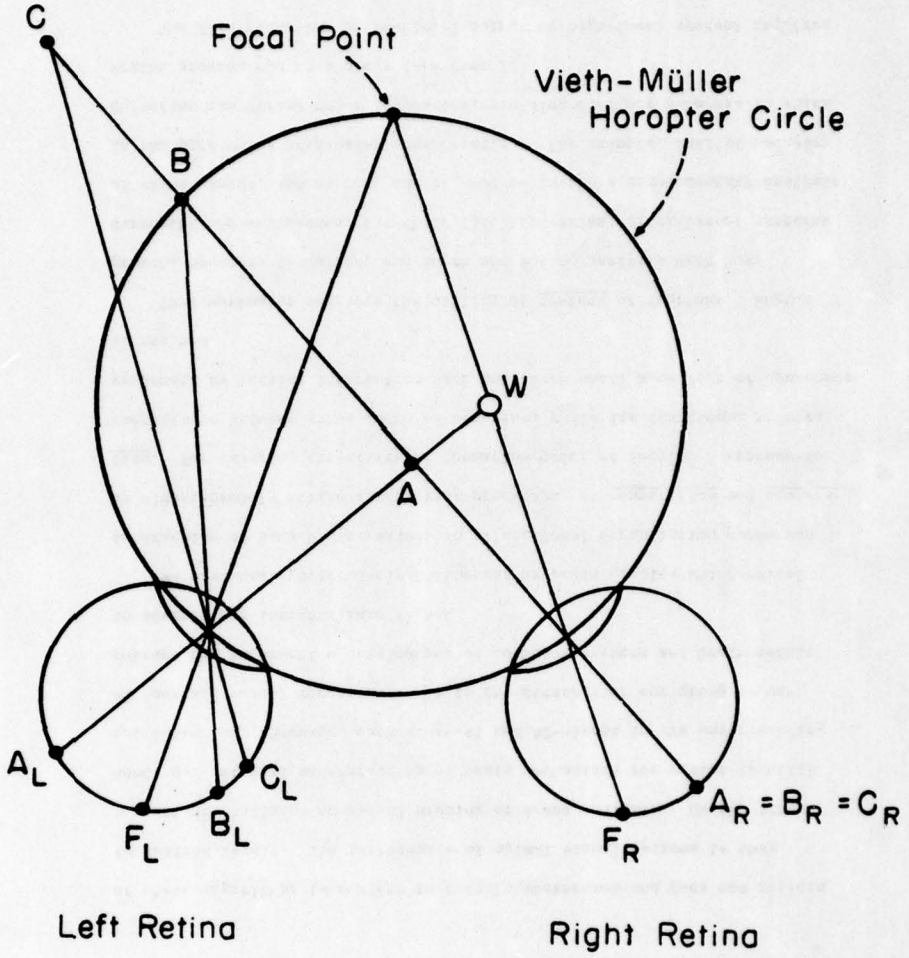



**Figure 1:** **The Notion of Disparity.**

**The points A, B, C with the same right retinal coordinate have disparate left retinal coordinates. It is this disparity that allows the inference of the depth of the stimulating point.**

Before leaving the topic of feature extraction, we should note that animal studies show that feature extractors can be 'tailored' by experience. In this way, the animal can quickly respond to those local features which occur significantly in his particular environment. The experimental analysis and theoretical working-out of such learning mechanisms is a major thrust of our group, but is beyond the scope of this paper--a review is provided by Arbib, Kilmer and Spinelli [1976].

Let us turn now to high-level systems. These employ the output of low-level systems to build the interpretation (internal model) of the visual input. This construction requires the use of various 'knowledge structures'. It should seem obvious to one who drives a car that the general view of a road is expected: the road elongated (with perspective distortion), cars, dotted lines, gravel on roadsides, sidewalks, grass, trees, and houses; they roughly fit together into a very familiar model, albeit one that possesses enormously many variations. Similarly, our prior experience of driving at night allows us to 'see' a red pinpoint of light as a car or traffic light. Moreover, we can use a few features to perceive an object from many different views, even with significant portions of an object occluded or obscured in shadow. Again, perception differs when we look for a house rather than simply driving down a street.

In this paper we shall outline a number of approaches to both segmentation and interpretation in which local processes communicate to achieve global organization. In Part II, we shall turn to detailed computer algorithms for segmentation of static scenes. Even the best



of these algorithms is subject to certain weaknesses and does not perform to desired levels. The performance of animal visual systems is thus a constant challenge to our AI studies of scene analysis. On the other hand, our detailed understanding of brain mechanisms for vision is still quite limited. However, each area--AI and BT--leads in the understanding of certain visual mechanisms. It is the strategy of our group to encourage the interactive development of computer systems and brain models to advance our insights into sight.

We have been particularly influenced by brain studies which reveal interaction of many brain regions in a functional organization subserved by simultaneously active interacting processes of competition and cooperation. For example, the reticular formation model of Section 3 represents competition between gross modes of behavior, while its functional modules cooperate by passing information back and forth until some form of consensus is reached.

This viewpoint suggests the utility of systems of routines. Region growth, boundary formation, and depth and motion analysis each have strengths and weaknesses; but their fidelity varies for different patterns of visual input, and so they can be used to invoke a complementary analysis in the particular area under consideration. For example, initial boundary detection can define major boundaries, yielding distinct subareas in which region growing can take place (see Part II).

The interpretation of low-level output by high-level systems requires the extensive use of 'general knowledge' to compensate for the relatively limited data supplied by visual input in a specific situation. It also must include routines for dealing with changes in perspective and distance, and with the occlusion of one object by another. This mention of 'general

knowledge' leads us to one of the central studies in AI at present: the organization, storage, retrieval and application of knowledge. (See Minsky [1975] and, for a useful collection of articles, Bobrow and Collins [1975].) The current view is that the elementary 'knowledge units' must be grouped into larger structures so that access may be gained from relevant 'entry points' elsewhere in the 'knowledge network' to avoid a great degree of redundant storage.

The low-level systems can also receive guidance from the high-level systems. Thus, the partial model, expected objects, context, direction of the light source, and so on, might all aid segmentation. Conversely, the low-level system can pass, in addition to the delineation of a set of regions, other useful information to the high-level. Regions in proximity, with similar hues but different intensity, signal a hypothesis of shadow and this information can be passed from low-level to high-level systems. Other hypotheses include reflections, different orientation of surfaces to light sources, as well as characteristics of the surfaces.

If the various routines are to be viewed as a system, then the operation and output of these subprocesses must be properly coordinated, including the determination of the consistency of results and the extent to which any particular result must be verified by companion processes. The coordination mechanism can be viewed as what computer scientists call an operating system with a set of resources and tasks. The subset of resources that will be employed at any moment is dependent upon the nature and importance of the particular subgoal to be achieved. The system must systematically fit the results of boundary analysis to the regions determined by depth, texture or motion analysis. If the results are inconsistent,



it must be able to direct a more detailed investigation to determine the subsystem and results which are in error and properly modify them.

Much of the work on image understanding grew out of earlier studies of pattern recognition--the study of techniques for classifying an isolated pattern, such as a handwritten numeral, a face, or a fingerprint. In much of image understanding, we stress the need for segmentation prior to object recognition. However, in some cases, local cues (e.g., the texture of a cat's fur) may trigger higher-level pattern recognition prior to the completion of segmentation. The properties of the classified object can then guide the segmentation process. We shall have more to say about the interaction of low-level and high-level systems in Section 3.

In much AI work, the process of scene analysis stops when the scene has been segmented, and each segment bears a name: 'tree', 'house', 'grass', 'sky', and so on, would be typical labels for the analysis of an outdoor scene. Workers in AI, and many psychologists interested in verbal behavior, stress the importance of this symbolic representation of the input (Newell and Simon [1976]). Of course, many problems for which a computer vision system would be desired would require the further specification of the relationships between objects. For example, even the simpler blocks-world systems found it necessary to go on to asserting objects, such as geometric position and support, in order to reason about assembly and disassembly of blocks-world structures. From an AI point of view, these relationships would also be represented symbolically.

However, the internal representation of an object must often include a program for the system's interaction with it. Arbib [1976] has introduced a new concept of a schema[1] which represents significant chunks of the world. with routines for recognizing the occurrence of an object or situation, for

[1] Arbib [1975a] provides some historical background on his theory of schemas; while Arbib [1975b] compares schemas with Minsky's concept of a frame as a 'knowledge unit'.

making use of context, and for acting appropriately. He posits a whole population of schemas of varying levels of activation, with parameters tuned to exigencies of the environment in such a way as to prepare a variety of action routines to guide the system in interacting with that environment. We suggest that the two views can be reconciled by regarding the symbolic representation of the world as an approximation to the full schema representation.

If we consider only the high-activity schemas, discarding all schemas whose activity is below a certain threshold and discarding all but the crudest information about the tuning of parameters, we may capture a rough description of the scene, structured by the answers to questions like 'What are you looking at?' (symbols corresponding to gross states of object motion) and 'What are you doing?' (symbols corresponding to action routines). We posit that the brain works with the full analogue representation, with the symbols remaining implicit in the pattern of schema activation unless elicited by a need for verbal interaction. However, in computer systems, it often proves cheaper to work with symbolic representations tailored to the mode of scene analysis involved, restricting the computation to the sequential exploration of a restricted set of representations at any one time. In the present paper, we shall concentrate on the stripped-down problem in which the action-routine of a schema is restricted to conveying the symbol associated with the object that the schema represents[1].

[1] For a look at the problem of parameter-tuning in the neural control of movement, see Section 2 of Arbib [1975a].



Let us briefly outline the main components of a schema:

(i) Input-matching routines: A schema corresponds to an 'object' whether a concrete object in the usual sense, or a concept at a very abstract level, such as 'winter', or 'a differential equation' or some social occasion. A schema thus requires routines whose job it is to search sensory stimuli, as well as messages from other schemas, for cues as to the correctness of the hypothesis that the 'object' which the schema represents is present in, or descriptive of, the system's environment. In their fullest generality, the input-matching routines will not simply match static aspects of the environment, but will match dynamic aspects--as, when crossing the road, rather than perceiving the make of the car, one is more interested in perceiving how the car is moving so that one can avoid it.

(ii) Action routines are available to guide the activity of the system in interacting with the 'object' which the schema represents. As input-matching routines adjust the parameters of the representation more accurately, the action routines should be adjusted so that the action they would release becomes more and more appropriate for the current environment and goal structures--as, in perceiving the motion of a car, we determine which way to jump to avoid it.

(iii) Competition and cooperation routines: Different schemas will compete to 'cover' the 'object' in a given part of the animal's world--is that moving object a nearby insect or a distant bird? At the same time, various schemas will cooperate to between them provide a coherent representation of a number of regions in the world--if we recognize one region as a face, it becomes more likely that the region below it is a body.

This process of competition and cooperation will depress some schemas and increasingly activate others, yielding a 'collage' of active schemas which provides an acceptable representation of the environment. Further competition and cooperation routines (planning routines) are then required to turn the range of possibilities afforded by the action routines of the activated schemas into a coherent plan of action for the organism. For example, if an animal sees a bowl of water and a dish of food, its state of hunger or thirst will help determine which way it turns.

As well as schemas for objects, we may also have more abstract schemas such as one for winter. Now at the change of seasons, the first fall of snow may be the signal for winter--so that we must posit the activity level of the snow-schema as providing excitatory input to the winter-schema. However, in the normal course of events, the organism knows that it is winter, and can use this contextual information to favor the hypothesis that a white expanse is snow rather than burnished sand, say, or moonlit water. It is this type of reciprocal activation (whether we regard it as an additional input, or as the action of a cooperation routine) that gives the system of schemas its heterarchical[1] character.

In an extended theory of schemas, one must not only spell out, for example, the detailed working of the competition and cooperation routines, but must also specify how input-matching actually serves to tune the action parameters; specify the way in which the matching of dynamic properties

[1] Strictly defined, a 'heterarchy' is a system of rule by alien leaders. But in AI, stimulated by Minsky's response to McCulloch [1949], it now denotes a structure in which a subsystem A may dominate a subsystem B at some other time.



of 'objects' enables the organism to act in a predictive fashion; and specify the way in which the organism can 'learn from experience'. This updating of memory structures must involve the combining of old schemas to form new schemas; the tuning and editing of schemas to better fit them to a changing world; and provision of increasingly rich relational information--embodied in part in competition and cooperation routines--to coordinate schemas to better encode relations between the 'objects' that they represent. The understanding of such procedures provides an outstanding challenge to both AI and BT. In this paper we shall devote most attention to various neural (Section 2) and computer (Part II) approaches to segmentation of the visual world. In Section 3, we further explore the notion of schemas, and give a simplified theory of their competition and cooperation, because it is these kind of structures that utilize the results of the segmentation processes.

## The VISIONS System

A more explicit feel for the role of high-level and low-level systems in scene analysis can be gained from schematic views of the VISIONS project (Hanson and Riseman [1974, 1975]), a computer-based visual perception system we are developing at the University of Massachusetts at Amherst. The structure of the entire VISIONS system, Figure 2, is quite complex and involves the interaction of many subsystems. In terms of our basic division, we may distinguish low-level processes whose goal is the segmentation of the image into regions and (major parts of) conceptual objects, each with a set of associated visual features; and high-level processes to construct a model of the three-dimensional world represented

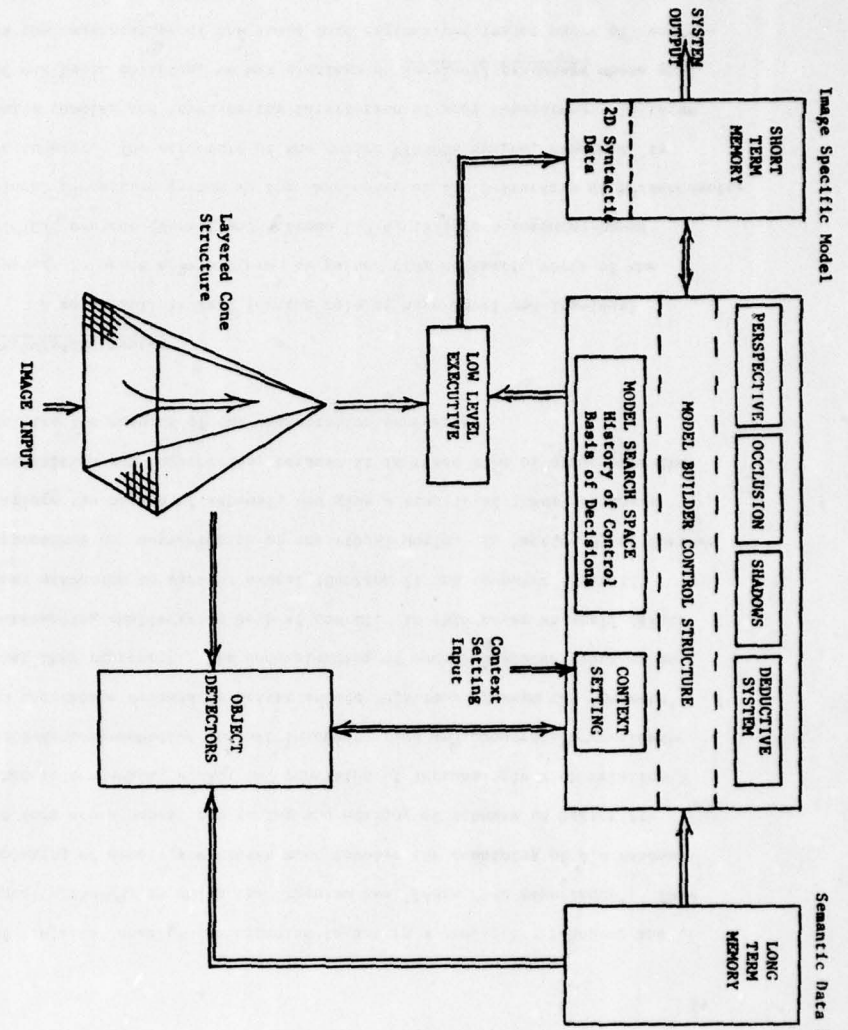



**Figure 2:** Global Organization of the VISIONS System.

The system divides into two major sections: a low-level system for segmentation of an image into regions representing (major parts of) the conceptual entities to be recognized and a variety of feature descriptors of these regions; and a high-level system whose goal is the interpretation of the image by building a three-dimensional semantic representation of the world depicted in the image. The low-level system is a hierarchically structured array of local processes (a processing cone) which transform and reduce the image data in parallel. The high-level system employs modular processes which construct a model by interfacing the symbolic output of regions and boundaries to stored world knowledge in a semantic data base.

in the scene. The latter involve the use of a semantic data base, expectations about the scene, deductive mechanisms, analyses of perspective, occlusion, and shadows, etc.

The design goals of the system include the utilization of:

1) a flexible interface of visual data to semantic knowledge;
2) flexibility of processing: data-directed, knowledge-directed and model-directed--with many kinds of knowledge, many levels of representation and redundancy of information;
3) a sketch of control history and basis of decisions to allow directed but limited backtracking;
4) an interface between local and global processing, as well as serial and paralllel processing.

The low-level system subdivides into two components: the cone structure and the low-level executive. The cone structure corresponds to the array of feature preprocessors we have seen to constitute the front end of animal visual systems. It is meant to provide a general computational structure for the numerical analysis of visual data. The processing cone is a parallel array computer that is hierarchically organized into a layered system (Figure 3): its major function is the transformation and reduction of the large amounts of data normally found in digitized images.

Information flow up, down, and laterally within the cone is controlled by defining local parallel functions, applied to local windows which are duplicated across the entire array. Functions operate on the $256^2$ grid of image data and reduce it, layer by layer, to single cells each of which contains information extracted from the entire scene. Many interesting parallel algorithms can be developed for processing of images;



e.g., algorithms for edges and lines, regions, texture, etc. For related work on hierarchical structures see Kelly [1971], Rosenfeld and Thurston [1971], Uhr [1972], Klinger and Dyer [1974] and Tanimoto and Pavlidis [1975].

Many of the procedures that we will detail in Part II are operations that can be applied in parallel to local windows of the scene. In the cone implementation, the results of these parallel operations are stored in pseudo-image arrays which are also available for further processing by local operators. During a reduction process upward through the layers in the cone, the data are reduced because portions of each window are non-overlapping. An iteration process allows the data to be analyzed and/or transformed at a fixed level of the cone; the size of the array remains constant due to overlapping of windows. A projection process allows information in upper layers to influence computation in lower layers.

The development of the model-building portions of VISIONS has proceeded under the assumption that a data structure, known as RSE (for Regions, line Segments, and Endpoints) has been successfully built as a result of low-level processes (Figure 4). Information in the RSE data structure is produced under the control of the low-level executive and is represented in symbolic form. The high-level processes accept this data as input and define the mapping of this information into progressively higher levels of organization[1].

[1] This general transformation of sensory data into different representations of increasing abstraction is similar to the structure found in HEARSAY, a computer speech-understanding system that we discuss further in Section 4.

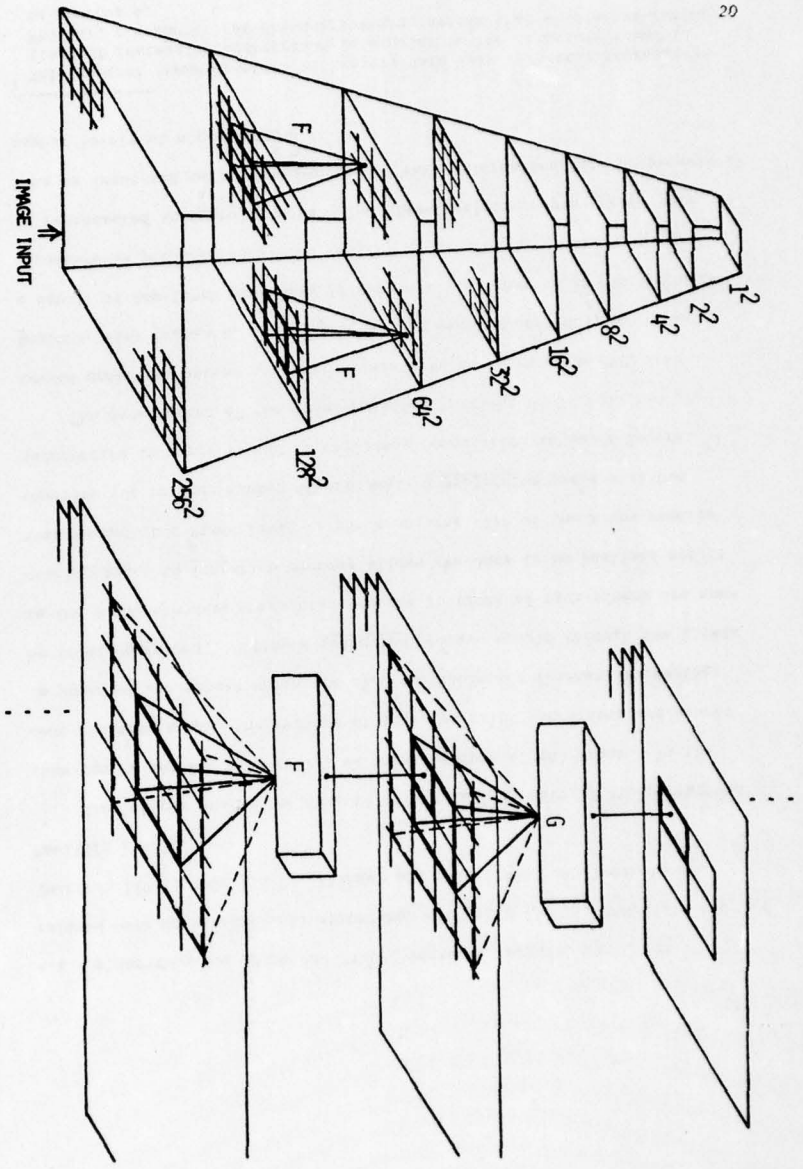


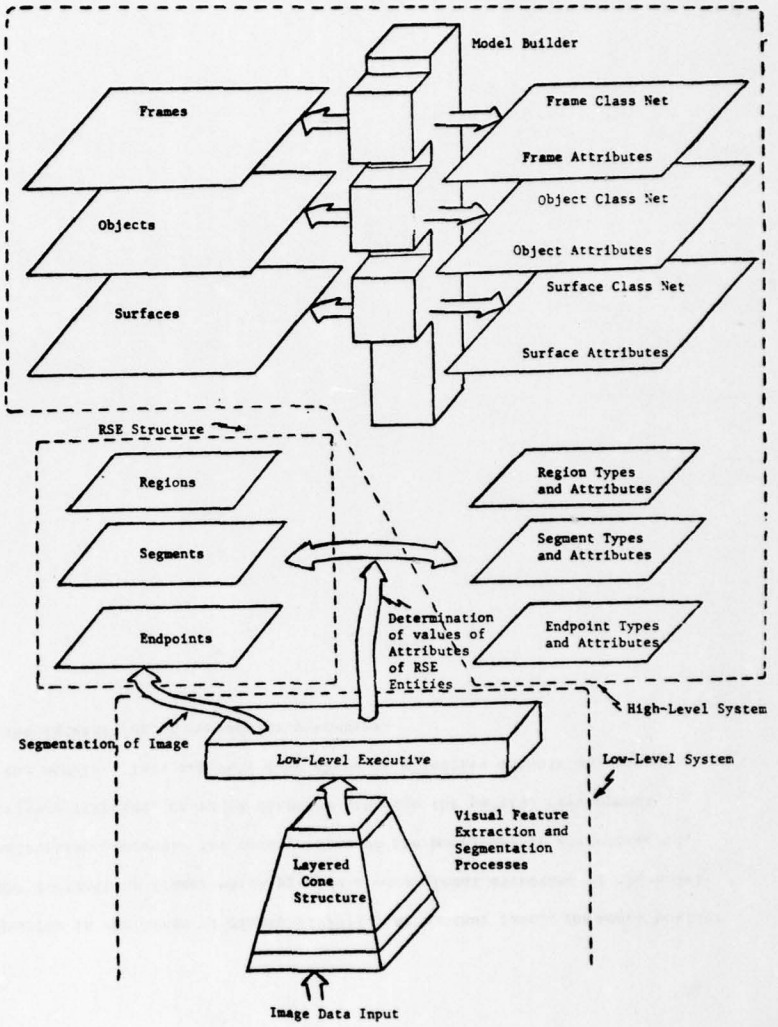

Figure 3: The Preprocessing Cone for the VISIONS System.

The processing cone is a simulation of a parallel array of micro-computers that is hierarchically organized into layers of decreasing resolution. A local operation (compacted on or between the layers of data) is defined in terms of a small window and is applied simultaneously across the entire layer. Information can flow: a) upward by operating on a window at layer i and storing the result at layer i + 1; b) laterally by operating on a window at layer i and storing the result in the center of the window at the same layer; c) downward by providing to an operation on a window at layer i, data from the unique parent of the cell at each higher level in the cone. General procedures for segmentation (e.g., line finders and region growers) and feature extraction (e.g., texture and color analysis) can be programmed as sequences of parallel algorithms.

Figure 4: Levels of Representation in VISIONS.

The segmented output of the cone represents two-dimensional visual syntactic information. It is stored in symbolic form as a layered graph in terms of regions delimited by line segments, and line segments delimited by endpoints. This data structure forms the major interface between the high- and low-level systems. The semantic levels are represented by surfaces, objects, and frames (or schemas) which are familiar submodels or scenarios, such as a road scene or sub-urban house scene. A node on each of these levels is defined in terms of the entities on levels below. The left hand set of planes represents the image specific model which is constructed by the model builder, while the right-hand set of planes represents a priori stored knowledge of the world. An image specific model is defined by pointers from the left side to the right side. Each of these pointers represents a hypothesis that a specific entity is a member of some class of entities that the system has knowledge of--for example a particular region $R_j$ may be pointed at by some object $O_i$ which has a pointer to the class of trees stored in long-term memory.





The goal of the VISIONS high-level system is the construction of a model which describes the major concepts and three-dimensional space of the scene under consideration. Again, this portion of the system may be divided into several subparts:

a) Image specific model or short term memory: this consists of the information contained in RSE and the interpretation placed on this data by the semantic processes, i.e., the left hand side of Figure 4 including RSE and the planes above.

b) World knowledge or long term memory: contains the general world knowledge which provides the data for structuring the image specific information into a model of the image, i.e., the right hand side of Figure 4 from the permanent attribute representation of RSE and up.

c) Control processes and sources of knowledge: these are the model building functions which are responsible for structuring the image-specific data, beginning with RSE and expectations concerning the image, into a consistent representation and interpretation of the image. They utilize modular knowledge sources of perspective, occlusion, shadow, deductive system, context setting, while maintaining and using a history of control.

At one level, the model builder is based on a 'hypothesize and test' mechanism employing many forms of knowledge: these include information permanently stored in the semantic data base ('long term' storage) and image-specific data (short term storage) in the form of the RSE data. The model will contain hypotheses about surfaces (and volumes), as well as schemas for objects and appropriate frames (representations of familiar



scenarios in the sense of Minsky [1975]). At another level, the model builder might instantiate frames which provide a more global direction to the model construction process. The construction of the model may be data-directed, knowledge-directed, or model-directed (through the partial development of the model). This may only take place by effective methods of control of the interaction of the modular processes.

## 2. An Approach to Segmentation

In Part II, we shall analyze in detail computational techniques for segmenting a scene--both by forming edges to provide boundaries which delimit edges, and by growing areas of a certain texture. A quick glance at the texture and outline of a tree suggests the complexity of systems which can handle natural outdoor scenes in full generality. In this section, we provide a simpler segmentation algorithm--derived from neural modelling--which handles an image which must be broken into regions, each labelled with a member of a small set of preassigned labels. This contrasts with the situation in Part II where the variety of color and texture is so immense that the system must generate a new set of labels appropriate to each scene.

### Segmentation on Depth

The general schema for segmentation on preassigned features (Arbib, Boylls and Dev [1974], Dev [1975]) can be motivated by the question of how disparity-detecting neurons (recall our discussion of Figure 1) might be connected to restrict ambiguities resulting from false correlations between pairs of retinal stimuli. Julesz [1971] invented random-dot stereograms to show, inter alia, that this depth perception can arise even in the absence of the cues provided by monocular perception of familiar objects. The slide for the left eye is prepared by simply filling in, completely at random, 50% of the squares of an array. The slide for the right eye is prepared by transforming the first slide by shifting sections of the original pattern some small distance (without changing the pattern



within the section) and otherwise leaving the overall pattern unchanged, save to fill in at random the squares thus left blank. When one slide of Julesz's arrays is presented to each eye, subjects start by perceiving visual 'noise' but eventually come to perceive the 'noise' as played out on surfaces at differing distances in space corresponding to the differing disparities of the noise patterns which constitute them.

Note well that both stimuli of the stereogram pair are random patterns. Interesting information is contained only in the correlations between the two--the fact that substantial regions of one slide are identical, save for their location, with regions of the other slide. The visual system is able to detect these correlations. If the correlations involve many regions of differing disparities, the subject may take some time to perceive so complex a stereogram--during which time the subjective reports will be of periods in which no change is perceived followed by the sudden emergence of yet another surface from the undifferentiated noise.

To clarify the ambiguity of disparity in Julesz stereograms, let us caricature the rectangular arrays by the linear arrays of Figure 5. The top line shows the 21 randomly generated 0's and 1's which constitute the 'left eye input', while the second line is the 'right eye input' obtained by displacing bits 7 through 13 two places left (so that the bit at position $i$ goes to position $i - 2$ for $7 \leq i \leq 13$) while the bits at position 12 and 13 thus left vacant are filled in at random (in this case, the new bits equal the old bits--an event with probability 1/4), with all other bits left unchanged.

The disparity array of Figure 5 suggests the stripped-down caricature of visual cortex which we shall use for our model. Rather than mimic a columnar organization, we segregate our mock cortex into layers, with the initial activity of a cell in position i of layer d corresponding to the presence or absence of a match for the activity of cell i of the right retina and cell i + d of the left retina. (This positioning of the elements aids our conceptualization. It is not the positioning of neurons that should be subject to experimental test, but rather the relationships that we shall posit between them.) As we see in Figure 5, the initial activity in these layers not only signals the 'true' correlations (A signals the central 'surface'; B and D signal the 'background'), but we also see 'spurious signals' (the clumps of activity at C and E in addition to the scattered 1's, resulting from the probability of 1/2 that a random pair of bits will agree) which obscure the 'true' correlations.

Segmentation on Preassigned Labels

Consider, now, any set of preassigned features--with one spatially coded array of detectors for each feature. We then have the following situation for the problem of segmentation of prewired features:

Conceptualization: 'Layers' of cells, one for each preassigned feature.

Principle: Segment activity into a small number of connected regions, each confined to a single layer.



Can we, then, interconnect the 'layers' in such a way that clearly defined segments will form? We might imagine (but only as a crude first approximation) the resultant array of activity as then providing suitable input for a higher-level pattern-recognition device which can in some sense recognize the three-dimensional object whose visible surfaces have been so clearly represented in the brain.

Arbib, Boylls and Dev [1974] provided a neurally plausible interconnection scheme which yields qualitatively appropriate behavior of the array: the essential idea is given by the rule that there be moderate local cross-excitation within a layer; and inhibition between layers which increases as the difference in feature increases. Let than $x_{di}(t)$ represent the activity of the cell in position i of layer d at time t (where we now let activity vary continuously between 0 and 1); and let $h(j)$ and $k(j)$ be functions of the form indicated by Figure 6. The effective input to cell di from cell d'i' is then $h(d - d')k(i - i')x_{d'i'}(t)$ which is positive if $i - i'$ is small (cells belong to 'nearby' features i and i') but is otherwise negative. The strength of the interaction decreases as d' gets further from d (cells are from widely different locations) and $h(d - d')$ decreases. Adding together the effects of all the other d'i' cells, the change of activity of cell di is given in our model by the equation

$$x_{di}(t + 1) = \sum_{d'} \sum_{i'} h(d - d')k(i - i')x_{d'i'}(t) + x_{di}(t_0) \qquad (1)$$

where it is understood that the sum 'saturates' at 0 and at 1.

What this scheme does is allow a clump of active cells in one layer to 'gang up' on cells with scattered activity in the same region but

Figure 5: Segmentation on Disparity Cues.

(a) The two upper rows yield the five rows below; with a 1 occurring every time the corresponding 'retinal inputs' match with the disparity that labels the row. The job of the segmentation routine is to suppress spurious regions like C and E, so that B, A and D are available to high-level systems.

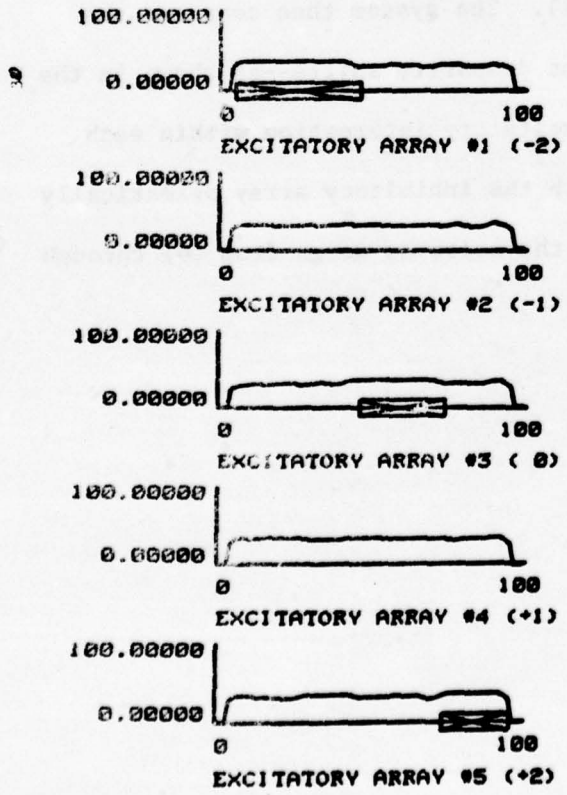
(b)-(d) Actual simulation of our model for segmentation on disparity cues. The right-hand side shows the input read in by touching a light pen to the disparity lines (compare the displacement terms in (a)). The system then computes the correlations for different disparity shifts--as shown in the 5 left-hand displays. Excitatory interaction within each array and interaction with the inhibitory array dramatically brings the signal out of the noise as we go from (b) through (c) to (d).

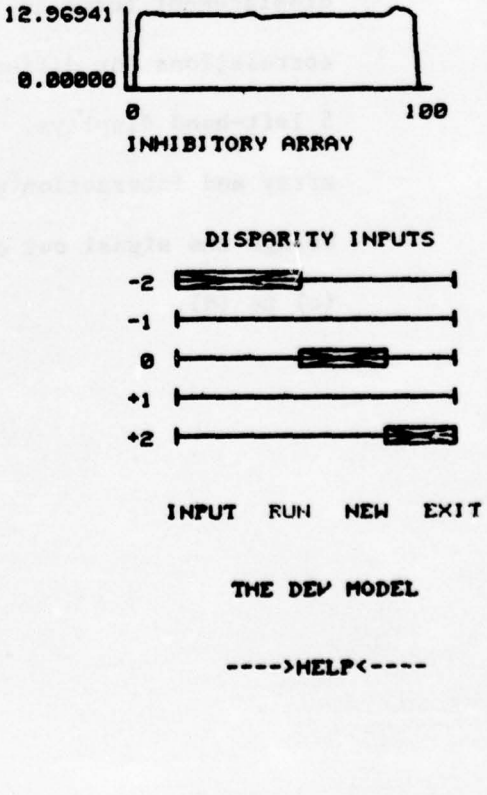
| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| and | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |

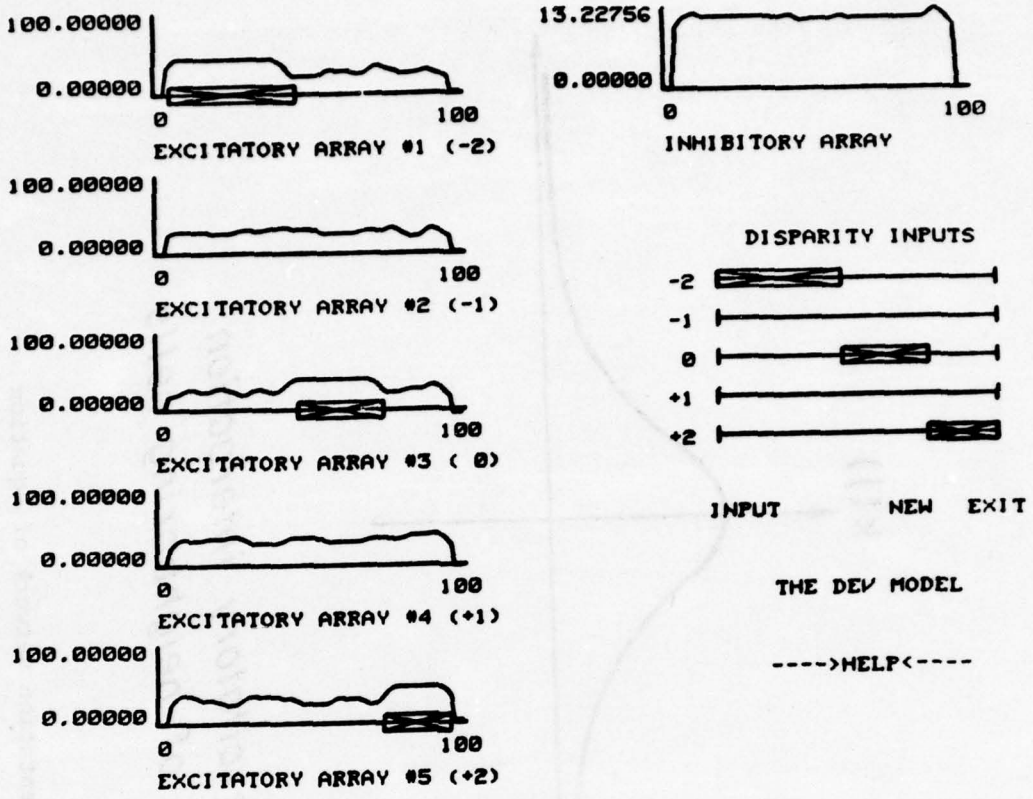
yield disparity array:

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| +2 | X | X | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| +1 | X | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| -1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | X |
| -2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | X | X |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |

A B C D E

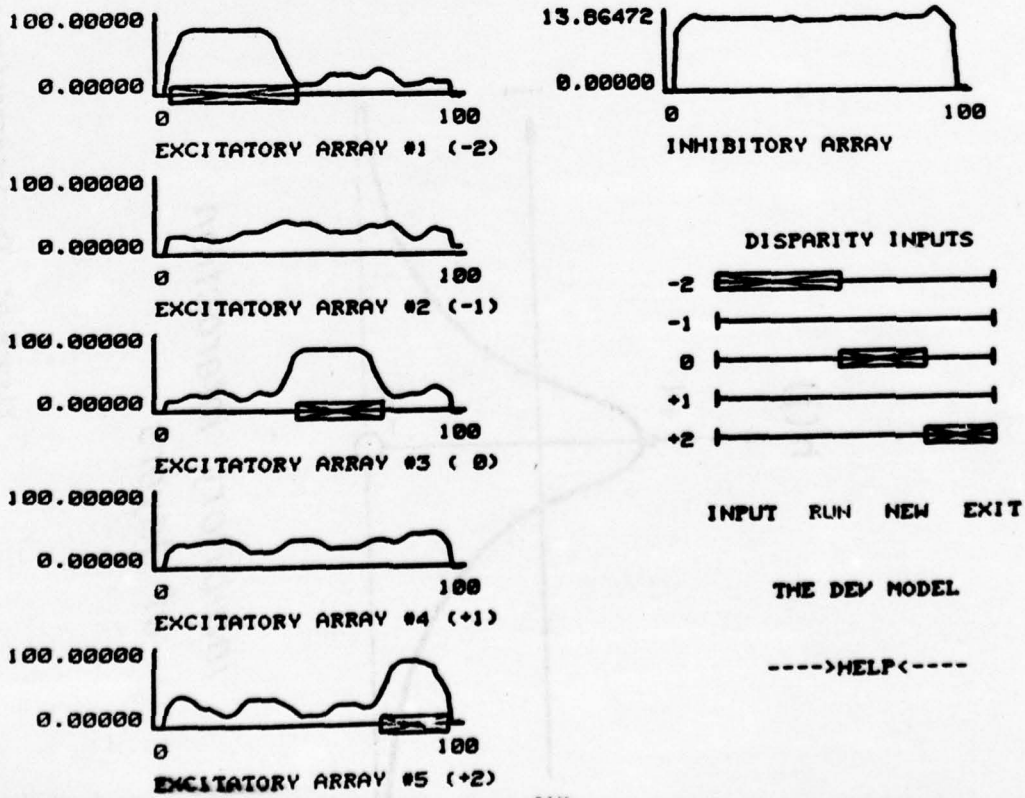


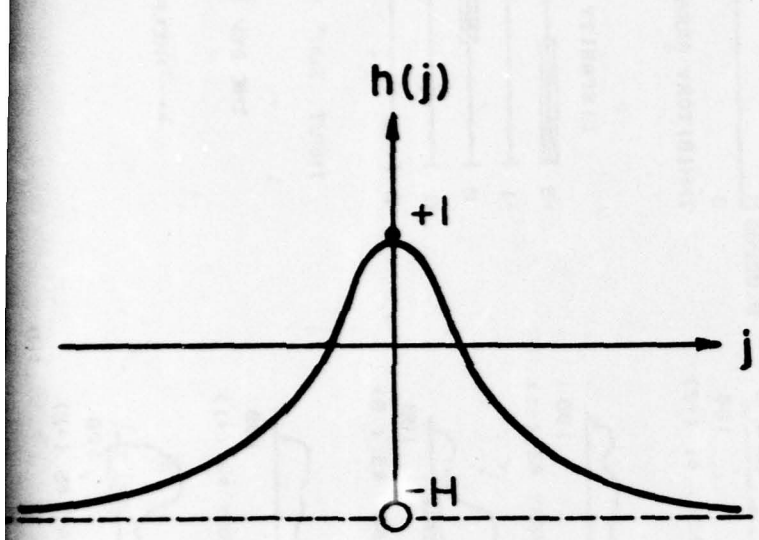

100.00000
0.00000
0
100
EXCITATORY ARRAY #1 (-2)
EXCITATORY ARRAY #2 (-1)
EXCITATORY ARRAY #3 ( 0)
EXCITATORY ARRAY #4 (+1)
EXCITATORY ARRAY #5 (+2)
13.22756
0.00000
INHIBITORY ARRAY
DISPARITY INPUTS
-2
-1
0
+1
+2
INPUT
NEW
EXIT
THE DEV MODEL
---->HELP<----

5(c)

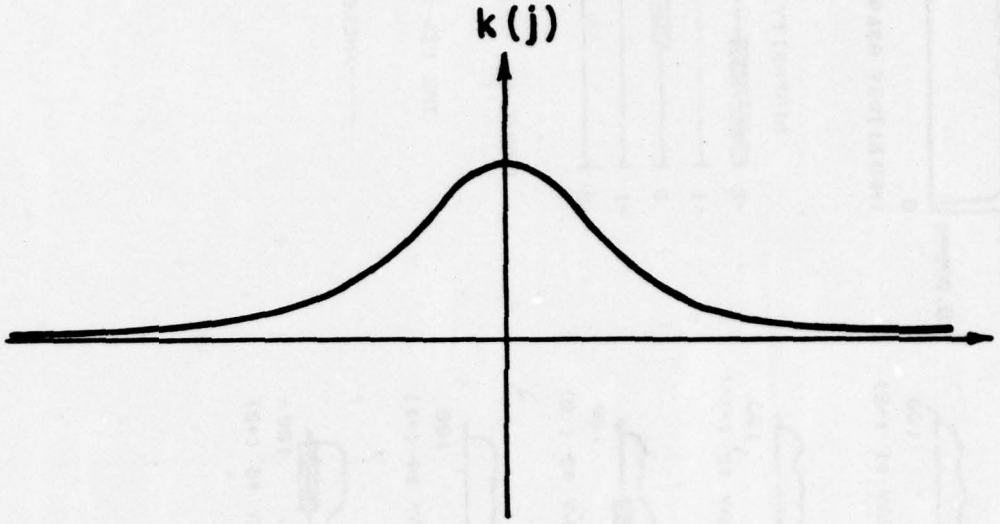



100.00000
0.00000
0
100
EXCITATORY ARRAY #1 (-2)
EXCITATORY ARRAY #2 (-1)
EXCITATORY ARRAY #3 ( 0)
EXCITATORY ARRAY #4 (+1)
EXCITATORY ARRAY #5 (+2)
13.86472
0.00000
INHIBITORY ARRAY
DISPARITY INPUTS
-2
-1
0
+1
+2
INPUT
RUN
NEW
EXIT
THE DEV MODEL
---->HELP<----

5(d)




*inhibitory interaction of layers*




*excitatory interaction of neighboring cells*

**Figure 6:** The interaction terms for the segmentation network of equation (1). h shows B excitatory interaction decreasing with increased difference in spatial location; k shows the inhibitory interaction for different feature points.

in other layers ($i \neq i'$), while at the same time recruiting moderately active cells which are nearby in their own layer (small $d - d'$). The system then tends to a condition in which the activity is clearly separated into regions, with each region having its own unique feature (layer of activity). In other words, such a scheme resolves feature ambiguity through suppression of scattered activity, thus permitting activity related to only one feature in any one locale. Moreover, returning to the stereopsis example, the dynamics of the model does represent the Julesz phenomenon of a noise stereogram taking some time to be perceived, with each new surface being perceived rather abruptly. This is simulated in the model by the fact that, once a sufficient number of clumps achieve high activity, the recruiting effect fills in the gaps between the clumps to form a good approximation to its final extent.

Although the analysis of moving scenes is beyond the scope of this paper, it is worth noting that Ross [1976] has extended Julesz's work to random process stereograms, and that Burt [1976] has extended the Dev model to handle moving stereoscopic input, as well as to explain certain phenomena of apparent motion. An important point of such a study is that moving boundaries provide a very powerful aid to segmentation. Thus, even though moving images offer far more data at the frame-by-frame level, they may--if properly preprocessed in terms of motion detectors--prove easier to segment than the static images on which we focus in Part II. Clearly, such a system leads to a representation in which the activated schemas are maintained for extended periods of time, with input serving to re-tune their parameters, with relatively little computation required for resegmentation and the activation of new schemas.



## 3. Cooperative Computation

Many contributions to brain theory suggest (Arbib [1975a]) that we model a set of brain regions as a cooperative computation system--a distributed structure in which each system has its own 'goal structure' for selecting information to act on from its environment, and for transmitting the results to suitable receivers (or simply broadcasting them). To this we must add that in many realistic systems, the overall behavior must be produced within a limited time and that there is often no right answer--rather the best answer available within the time limitations. Conventional computer science views debugging a program as rewriting it to remove errors in the sense of departures from an overall prespecification of its behavior. However, when the desired behavior of a system cannot be completely prespecified we do not restructure the system to get the 'right' answer. Instead, we need a system that can be continually restructured to ensure that over time the overall behavior of the system improves cumulatively.

The main technique of neurology is to learn about the structure of a system by observing lesion-induced defects. [Neurology is the clinical study of brain damaged humans. It is neurophysiology, together with related anatomical and chemical tools, that traces the behavior of single neurons and of neural circuits. Unfortunately, the gap between the region-by-region analysis of the neurologist and the few-neurons-at-a-time analysis of the neurophysiologist remains immense. While the neural net modeller, as in the Dev [1975] study, investigates the behavior of nets of formal neurons, the neurological modeller must study the effects of subsystem deletion in a distributed structure of cooperating systems. In many such systems, computation will proceed to some sort of completion despite damage--so long as the deleted subsystem is not an output system. To see this, consider the effect of removing one physician from a panel of physicians (each with his own strengths) trying to reach consensus on the diagnosis of a patient. In some happy cases, convergence on consensus may be sped up because the missing physician would have been

grossly wrong in his initial diagnosis had he been present. In general, however, the absence of a member from a panel will slow convergence--the other physicians will still be able to reach their decision, but could certainly have profited from his input. Finally, the results may be disastrous when the patient's disease is so unusual that only a specialist has any chance of correct diagnosis, and it is that specialist who is missing.

Clearly, then, each subsystem must have a good enough model of the others to communicate effectively to get some ability to make up deletions. This model can be a very crude one. For example, our model of the international banking system is simply that if we buy travelers' checks in one country, then we can spend the money in another country. Further information about the actual pattern of transactions required to bring this about is not going to affect our behavior.

This last observation probably offers the key to the case against executive control in the implementation of the cooperative computation strategy. In executive control, we do not require subsystems to know anything about other subsystems--it is the job of the executive to switch in a subsystem as and when it is needed on the basis of executive monitoring of the completion of other tasks. However, if each system is truly complex, a central controller could be overloaded by simulating or in other ways studying each system in sufficient detail to determine the communication scheduling. While the throughput of local communication strategies may be suboptimal, the time lost in suboptimal computation may be far less than that required by an executive to actually compute the optimum.



HEARSAY

Our discussion of VISIONS made it clear that such a style of cooperative computation not only provides the proper perspective for the analysis of brain function, but should also provide the proper way of structuring computer perceptual systems. Before turning to a study of cooperative computation, we look at an AI system which distributes the task of speech understanding between 10 or so separate processors. The HEARSAY group (see, e.g., Erman and Lesser [1975]) at Carnegie-Mellon University have built a speech understanding system which is to be run on a network of PDP 11's. At the time of this writing HEARSAY is still evolving. There are sufficiently many implementation difficulties in their approach to render premature an uncritical acceptance of their approach, but the general viewpoint is an intriguing one.

Each computer contains a knowledge source. One source is 'expert' at going from formants to phonemes, another expert at taking a string of words and making a syntactic analysis, while another can take partially analyzed sentences and look for semantic interpretations. One key idea of their system is that effective performance should not be dependent upon an executive controller which schedules tasks amongst these subsystems. Rather, there is a central communication center called a blackboard. Each knowledge source can take data from the level in which it is interested and work upon it, either returning to the blackboard an interpretation which wipes that question off the blackboard, or adding new questions. A basic analyzer might take certain phonemes off the board, and return a number of consistent words. A syntactic analyzer might issue a request to check whether a particular word it requires is present at a particular place in the input. It is clear that this is very much in the spirit of cooperative computation.

in the sense in which we have introduced it.

The uniformity of the structure and communication of these modular processes is a strong advantage for building a complex system--knowledge sources can be very easily added, replaced, or deleted without major side effects or system redesign. However, the generality of this approach as a system tool does lead to inefficiency when the designer wants to have one module send a message directly to another. The flexibility of more explicit and direct communication would avoid the overhead of writing and reading from the blackboard when the communication path is inherently narrow.

One deficit of the HEARSAY implementation is that changes on the blackboard can destroy 'history' which may be needed if backup is required when new data call for reinterpretation of other data. The basis of previous decisions is not accessible, only the results of the processing. This causes no difficulty when the hypotheses are correct, or sufficient redundancy is available via other processes, but it is not yet known how frequently these conditions may be achieved. The VISIONS group (which Lesser will join in early 1977) is thus experimenting with the use of a context tree to help direct backtracking. A problem there is to find the right interactions to link competing and consistent models.

The goal of reliable and effective performance of both VISIONS and HEARSAY is crucially dependent upon the quality of the initial stages of processing of the sensory data. The history of the field of pattern recognition has made it quite clear that high quality decisions cannot be based on low quality and unreliable sensory features. In HEARSAY, acoustic and phonemic processing must usually produce a small set of alternatives which includes the correct hypotheses. Otherwise, as the system works on higher levels of representation in the hypotheses of words, grammatical structure, and meaning, they will be based upon a somewhat deaf ear and will probably be incorrect. Similarly, in Part II of this paper, we examine the initial segmentation processes of VISIONS and visual systems in general.

Crucial to the general application of the HEARSAY approach is the choice of appropriate levels for data on the blackboard. These provide the true medium of communication, and so the use of a single blackboard



is probably a misleading feature. In fact, the HEARSAY implementation lets one processor lock out the others when accessing the blackboard--and this can create deadlock problems. The provision of greater simultaneity is not only more brain-like, but should lead to improved computer structures. Of course, communication channels do not guarantee that the system will converge upon a consistent representation of the input. The designer must anticipate conflicting interpretations and provide the means for resolving disagreements via locally executable rules. A challenging research problem is to determine whether this can be done without strong executive scheduling. The simplest form (although not very satisfying) would allow process A to always dominate process B during conflict if A is prejudged to be clearly more reliable than process B. Often it would be hard to estimate reliability, and a better strategy would demand further analysis by each subsystem to weight the confidences of each of the analyses, allowing the resolution of conflicts to be data-driven.

As the cost of microcomputers plunges well below $100 one can expect an increasing interest in computer architecture based on networks of minicomputers and microcomputers. It must be admitted that there are many problems attendant upon the current HEARSAY implementation. We must develop a methodology for implementation of such systems with complex local interactions; these problems overlap the current dialogue in the AI community with respect to schemas and frames and other kinds of representations of knowledge packets.

It is difficult for human designers to trace the processing and debug a large system with many local interactions. If, in fact, these

problems can be shown due to the novelty of implementing a complex computation on a minicomputer network, rather than to any inherent limitation in the approach itself, then cooperative computation should become ever more important in computer science. It will certainly provide an intriguing framework for the brain theorist. In any case, we argue the need for greater simultaneity in, and a finer grain of, cooperative computation than that offered by HEARSAY. To get an example of such fine grain computation, we now turn to a model of how schemas might interact in the interpretation of a segmented image.

## Competition and Cooperation Between Schemas

Imagine that a segmentation program has divided a scene into regions such as those shown atop Figure 7. With only this much information available, two quite different pairs of schemas may be activated to cover this input: In the first interpretation, the schemas would represent green ice-cream and a brown ice-cream cone. In the second interpretation, the schemas would represent the foliage and trunk of a tree. There would be competition between the pairs, and cooperation between the schemas within each pair. Thus the system of interactions shown in the lower half of Figure 7 would have two large attractors corresponding to the two natural interpretations, and very small attractors for the 'unreal' pairings--though these could be forced by a trick photograph or a Magritte painting. However (Figure 8) an initial configuration in which the schemas for foliage, trunk, ice-cream and cone have comparable activity will rapidly converge to a state of high activity in foliage and trunk schemas and low activity in ice-cream and cone schemas if context is introduced



by giving the grass schema a higher activity level than the hand schema. (Further illustrative visual examples of context significantly affecting local interpretation can be found in Baird & Kelley [1974].)

Casual inspection of a real-world scene frequently leads to ambiguities. These can be resolved--in the sense of having only one schema highly active for each region--either by shifting attention in a search for decisive context (as in Figure 8), or by closer scrutinizing of a region (as in looking for details of texture in the upper region which are more typical of foliage than of ice-cream). Thus, many schemas will have action-routines which direct attention to confirming or disconfirming features of the environment. Didday and Arbib [1975] have used this idea to analyze the series of parallel computations which direct human eye movements during visual perception. In the same way, any computer system will have limited resources, and will need high-level systems to aid the type and locus of low-level processing which is to be applied at any time.

Let us now try to put competition and cooperation between schemas on a formal level. We shall adopt an approach, due to Rosenfeld, Hummel and Zucker [1976], which uses a nonlinear iteration process (called a relaxation process) which operates on assignments of probabilities to the different labelling hypotheses for each region. This is a generalization of the constraint satisfaction method employed by Waltz [1975]. Since we may suddenly perceive the nature of an object purely on the basis of its context, the algorithm allows continuously varying weights to be assigned to each label for each region, and uses an operation which can increase, as well as decrease, those weights.

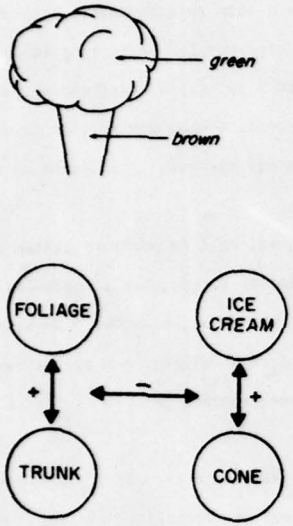



Figure 7: The 'tree' and 'ice cream cone' hypotheses compete for the input picture.



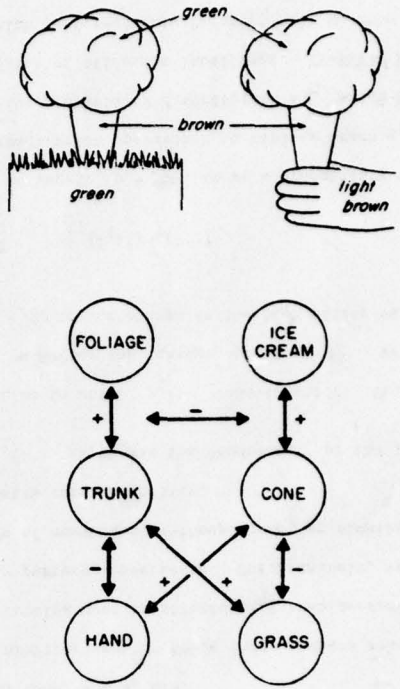



Figure 8: If the input is ambiguous as in Figure 7, we cannot expect rapid convergence to a state in which one consistent set of hypotheses overwhelms the other. However if a nearby region has high activity in a schema consistent with only one set of hypotheses, rapid convergence can follow. For example, if a region is clearly a hand, the interaction quickly enhances *'ice cream'* and *'cone'* activity, with resultant diminution of 'foliage' and 'trunk' activity.

Before going further it is worth stressing (Arbib [1976]) that the essential logic of this scheme was earlier used in the S-RETIC model of Kilmer, McCulloch and Blum [1969]. However, the work of Rosenfeld et al. grew directly from the need to augment the Waltz model by the use of weights, rather than from the RETIC model. We have here a case of convergent evolution. The S-RETIC was a model of the reticular formation of the vertebrate brainstem, and showed how cooperative computation could lead a collection of modules with differing samples of an input to converge to a consensus on the proper mode of activity of the animal. The main difference in the approach of Rosenfeld et al. is that convergence is towards a labelling, rather than towards consensus on a single mode.

We have a set $A = \{a_1,\ldots,a_n\}$ of regions, and a set $\Lambda = \{\Lambda_1,\ldots,\Lambda_m\}$ of labels. (Recall that we are now looking simply at labels for our schemas, rather than setting parameters for action routines.) A labelling $p = (p_1,\ldots,p_n)$ is a sequence of probability vectors $p_i: \Lambda \longrightarrow [0,1]$, with $p_i(\lambda)$ being the weight assigned by p to the hypothesis that $\lambda$ is the correct label for $a_i$.

We wish to design an operator F which--in the style suggested by Figure 7--will on iterated application move p towards a 'correct' labelling. The key idea is that the probability $p_i(\lambda)$ of a given label for $a_i$ should be increased (decreased) by F if other objects that have high probability labels are highly compatible (incompatible) with $\lambda$ at $a_i$. Thus the model uses compatibility functions

$$r_{ij}: \Lambda \times \Lambda \longrightarrow [-1,1]$$

such that if $\lambda'$ on $a_j$ frequently co-occurs with (or lends support to) $\lambda$ on $a_i$, then $r_{ij}(\lambda,\lambda')$ is positive; if their co-occurrence is implausible, $r_{ij}(\lambda,\lambda')$ is negative; and if their occurrences are independent, $r_{ij}(\lambda,\lambda') = 0$.



It is clear that the memory structures required to produce the compatibility functions may be quite elaborate. Returning to our example of Figure 7, the system would have to use the observation that regions 1 and 2 were contiguous, with region 1 above region 2, to obtain estimates like

$r_{12}$(foliage, trunk) = 0.7

$r_{12}$(foliage, cone) = -0.8

$r_{12}$(ice-cream, cone) = 0.9, etc.

Incidentally, the very arbitrariness of these three numbers makes it clear that the F we are constructing must be structurally stable--small changes in the $r_{ij}$'s must rarely perturb convergence. Unfortunately, we do not yet have rigorous proofs of convergence--though computer simulations are encouraging--let alone structural stability.

Each $\sum_{\lambda'} r_{ij}(\lambda,\lambda')p_j(\lambda')$ expresses the 'consensus' of the labelling of $a_j$ by p as to the direction in which $p_i(\lambda)$ should shift. Thus, to satisfy the 'key idea', we define the 'change operator' $\mathcal{K}$, which gives the first approximation $\mathcal{K}p$ to the change in the probability vector p, by

$$(\mathcal{K}p)_i(\lambda) = \sum_j [\sum_{\lambda'} r_{ij}(\lambda,\lambda')p_j(\lambda')]$$

There is no reason to expect $p + \mathcal{K}p$ to be a probability vector. Therefore, we need a normalization operator R to replace each $q_i$ of a vector q by a corresponding probability distribution $Rq_i$ which better reflects the compatibilities of different labellings. (S-RETIC had a sequence of operators which performed this R-function.) We then define our updating of the vector p to be the new probability vector

$$Fp = R[p + \mathcal{K}p]$$

We imagine the following operation of this procedure:

(1) The segmentation routines divide the original data into regions. Shape and texture descriptors are used to assign initial probabilities $p_i(\lambda)$ to appropriate labels $\lambda$ for each region $a_i$.

(2) Information such as relative position and the nature of the boundary would be used to generate the compatibility coefficients $r_{ij}$ (cf. Yakimovsky and Feldman [1973]).

(3) F would be iterated a dozen times, say, to provide enhanced probabilities. If the result is ambiguous, interaction with other systems--perhaps using higher-level context more subtle than that expressible in the $r_{ij}$ (e.g., familiar submodels with consistent parts should reinforce proper labels)--could be involved in disambiguation, with the possibility of reinitiating F using a new set of probabilities.

In Section 2, we discussed an algorithm (Arbib, Boylls and Dev [1974], Dev [1975]) which falls into the general 'relaxation procedure' paradigm that allow 'clusters' of low-level features having compatible labels to reinforce one another in segmentation processes. Interestingly, this algorithm was developed independently of the rise of relaxation procedures in AI.

We have already noted the similarity of the nonlinear probabilistic model to the S-RETIC, but with the emphasis on 'proper labelling' rather than on 'mode consensus'. The relationship of S-RETIC to Dev's and other brain models has been discussed by Montalvo [1975].



Semantics and Segmentation

Tenenbaum and Barrow [1976] bring interpretation and semantics right down to the heart of the segmentation process. In an application of the relaxation procedure that we have just described, they integrate the approaches via constraint satisfaction. This technique is essentially a relaxation process without probabilities for the compatability coefficients. Relationships between regions are used to constrain the possible labels of one region given a set of labels for the other region. For example if one region is labelled lake, then the region below should have the label of sky (if it exists) removed from its list of labels. In this manner, spatial, functional, and other semantic relationships between a given region and all adjacent regions can be listed to restrict the interpretation of that region.

If labels for possible interpretations are automatically produced for all elementary regions (even individual picture elements!), then the local constraints in different areas of the image can propagate towards the goal of finding one or more globally consistent interpretations of the scene. Note that if there are no semantic constraints, then unguided segmentation by standard region growing techniques (Part II) takes place. In interpretation-guided segmentation, pairs of adjacent regions with the same unique interpretation are 'safely' merged first; while merges with disjoint sets of labels are never allowed. Otherwise low contrast boundaries are merged and the new set of labels is formed by intersecting the labels for the original region. The intersection is further restricted by demanding consistency of each of these labels with the attributes of the newly-formed region; if the new set of

labels is empty, then the merge is blocked. Thus, after each merge, reinterpretation takes place by propagating constraints until there are no changes and then the next safest merge takes place.

Using this approach semantic constraints via manually supplied labels, geometric models, maps, and partial descriptions from other scene analysis programs are possible. The authors demonstrated the interpretation of a simple office scene (door, wall, baseboard, picture, doorknob) by restricting the label set of a one pixel region to be the unique correct interpretation. The process was highly effective in carrying out 214 safe merges, 43 unsafe merges, resulting in a unique consistent interpretation for the final 11 regions.

These impressive results must be viewed with caution. The domain was very limited with few objects, so that it was feasible to allow local areas to take on all possible interpretations. The crucial point is that with a much larger number of objects, one needs techniques which can limit the number of labels for each area. The degree to which one can satisfactorily label the possible interpretations of a small section of an object on the basis of purely local information is still uncertain. It would appear that the technique would be more effective after a symbolic representation of large regions and boundaries has been extracted. Then cues of occlusion, perspective, shadowing as well as semantic labels can be used in their constraint-satisfaction scheme.

The potential power of semantics in guiding segmentation is predicated on the ability to define meaningful constraining relationships between objects. Many of those used by Tenenbaum and Barrow are basically 2D spatial relationships, and since this simple scene involved only the wall plane (i.e., 2D), the relationships were sufficient to achieve the



desired goals. With more general three-dimensional scenes which might be viewed from many different perspectives and involve occluding objects, there are few 2D spatial representations that remain invariant. This requires three-dimensional representations of shapes of objects and relationships between objects, but this problem is non-trivial. It can be expected that the data base will contain far more instances of likely relationships (tree crown atop trunk) than of the unlimited set of unlikely relationships (such as ice-cream atop tree trunk!). The presence of objects and their relationships might be optional or variable (e.g., the presence of guard rails alongside a road). In the 3D world the size of a region in an image does not constrain the label of an object without consideration of the distance of the object.

Earlier, we noted that pattern recognition procedures may sometimes be triggered prior to segmentation. One can provide procedural knowledge (i.e., knowledge stored as algorithms which can be applied to the situation at hand) in the form of pattern recognition programs which compare expected features of objects and the actual features of a region in order to provide coarse likelihoods for the alternative identities of a region. In Part II of this paper, we will give examples of histograms of features of various objects such as sky, tree crown, grass, roof, etc. Each seems to have a 'signature' in terms of its 'waveform' or distribution of feature values. This implies that the object recognition routines might be very helpful during the segmentation of the more difficult areas of the image. Given that the problem has proven intractable to so many previous attempts, one should consider the ways to relax the demands for complete segmentation prior to initial labelling of regions; these processes might interact

and communicate. Trees in sunlight have a wide bandwidth in hue and intensity. Consequently, they can provide aid to a region and texture analysis in a particular area. Of course there might really be several regions (possibly textured) whose union gives HSI distributions similar to the tree's. The low-level system must manage these problems, only accepting suggestions and not conclusions from individual subprocesses. This management of diverse processes fits in naturally with a need for directing subsets of object routines to particular regions of the image. Again they could be controlled by a low-level executive or one could structure feature cues to invoke them in a heterarchical fashion, but in either case the cooperation of these processes would aid both low and high-level types of analysis.

The question facing AI reserachers is one of intuition concerning the payoff in attempting to structure the syntactic-semantic tradeoff in a machine. We expect the tradeoff to be task-dependent, and vary from system to system. At one extreme is the attempt to avoid semantics entirely in early visual processing and at the other is its complete integration in early visual processing. The cat visual system simply enhances contrast at the retinal level, while the frog has retinal 'bug detectors'. Again, an assembly-line robot has simply to choose between a small set of alternatives, while analysis of outdoor scenes continually provides challenging novelties. If the shadow areas in a road scene at night are to be interpreted, one cannot rely on segmentation processes to set proper thresholds for extracting meaningful boundaries. However, we do believe in the desirability of some degree of segmentation prior to the introduction of semantics. One must have a vision system which is initially



data driven whether one is viewing a scene of sense or nonsense; otherwise the system's expectations of what is out there might be nothing more than day-dreaming. Once structures for partial segmentation are available, feedback loops between high and low-level should allow both to cooperate and update each other in a dynamic fashion.

## 4. An Overview

Our view of the overall design of a visual system can be succinctly summarized in Figure 9. The overall goal of the system is to infer from the image an interpretation of the scene--which may name and locate certain objects in the scene, or provide programs to guide interaction with the environment of which the scene is a sample. Eyes or cameras form an intensity map of a scene. The map will often cover only part of the scene, and it will be limited in resolution.

The light intensity at a point is almost meaningless. Processes are thus required to extract features from local 'windows' (local in space and, possibly, time). These are features which can provide meaningful cues to the interpretation process. Features may combine two simultaneous images to provide depth cues, or may combine several successive images to provide motion cues. Other features will relate to presence of edges, local texture, and so on.

The multiplicity and overlap of windows, and the inherent ambiguity of local samples, implies that the output (A) of the initial feature extraction will contain much spurious activity. Thus processes--like those we studied in Section 2 and shall meet again in Part II--are required to eliminate false disparity cues, multiple indications of edges, and so on. The resultant 'cleaned up' feature map, (B), provides an array of features which is far more likely to aid the interpretation process than the original intensity map. It thus appears to correspond to the 'primal sketch' of Marr [1975]. This representation is 'quasisymbolic' in that the purely numerical assignment of intensity values in the original



map has been replaced by an assignment to each point of feature descriptors together with intensity levels which evaluate the extent to which the feature occurs in a neighborhood of that point.

In some cases, local features may directly activate schemas to interpret a region of image. However, in most cases, further processes of aggregation (to be discussed in Part II) are required. Edge features are to be aggregated into lines and curves which may delimit regions. Regions can also be formed by aggregating local areas which--at least in the context of the specific image--have key features in common. The resultant segmented image, (C), provides the major input to the semantic, high-level processes. It contains descriptions both of regional properties, and of relations between regions.

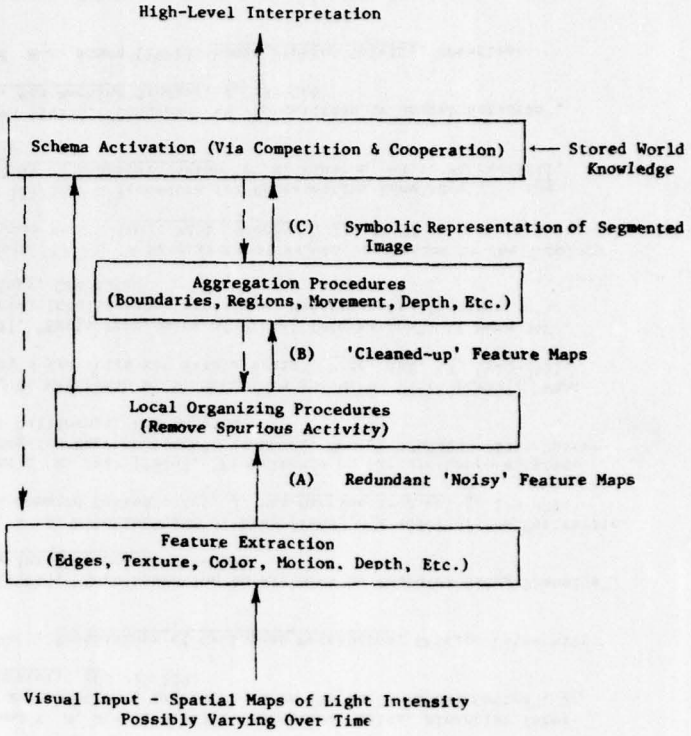
High-level processes can then act upon (B) and (C) to interpret the scene. Schemas can be activated by local properties of appropriate regions: and by global considerations--by competition and cooperation with other schemas, and by use of the relationships in (C) on the basis of world knowledge.

The flow of information is by no means one-way--the dashed arrows give a sample of some of the more important feedback paths. It is important to note that the environment will frequently contain more data than the system can handle at any one time. We thus require mechanisms for the direction of attention--allowing the system to focus on different places, or at different levels of resolution, in the image--in response to the ongoing needs of the process of interpretation. Finally, we note that almost all aspects of the system--from feature extraction to the organization of control structures for world knowledge--are subject to learning.

Figure 9: The Overall Design of a Visual System.

The visual input over space and time goes through a series of transformations until a satisfactory high level interpretation is achieved. Feature extraction over local windows (in space and time) is followed by simple operations which allow local organizing processes to remove spurious activity due to such things as overlap of windows and false disparity cues. At the next level processes for aggregating local edges and areas into boundaries and regions are applied. This representation is at the symbolic level; the boundaries and regions have names and lists of descriptive attributes for texture, color, depth, etc. Schemas are activated by local properties and interact in forming an interpretation of the scene. Important feedback paths are shown as dashed arrows.





## References


M. A. Arbib [1975a], "Artificial Intelligence and Brain Theory: Unities and Diversities", Annals of Biomedical Engineering, 3, 238-274.

M. A. Arbib [1975b], "Parallelism, Slides, Schemas, and Frames", in Two Papers on Schemas and Frames, COINS Technical Report 75C-9, Department of Computer & Information Science, University of Massachusetts, Amherst.

M. A. Arbib [1976], "Segmentation, Schemas, and Cooperative Computation", in Studies in Biomathematics (a volume in the MAA Studies in Mathematics) (S. Levin, ed.), in press.

M. A. Arbib, C. C. Boylls and P. Dev [1974], "Neural Models of Spatial Perception and the Control of Movement", in Kybernetik and Bionik/ Cybernetics and Bionics (W. D. Keidel, W. Handler, M. Spreng, eds.), R. Oldenbourg, 216-231.

M. A. Arbib, W. L. Kilmer and D. N. Spinelli [1976], "Neural Models and Memory", in Neural Mechanisms of Learning and Memory (M. R. Rosenzweig, E. L. Bennett, eds.), MIT Press, 109-132.

M. L. Baird and M. D. Kelly [1974], "A Paradigm for Semantic Picture Recognition", Pattern Recognition Journal, June, 6, 61-74.

H. B. Barlow, C. Blakemore and J. D. Pettigrew [1967], "The Neural Mechanism of Binocular Depth Discrimination", J. Physiology, 193, 327-342.

J. Beck [1975], "Surface Color Perception", Scientific American, August, 233, 82.

D. G. Bobrow and A. Collins [1975] (eds.), Representation and Understanding: Studies in Cognitive Science, Academic Press.

P. Burt [1976], Ph.D. Thesis, Computer and Information Science, University of Massachusetts, Amherst.

P. Dev [1975], "Computer Simulation of a Dynamic Visual Perception Model", Int. J. Man-Machine Studies, 7, 511-528.

R. L. Didday and M. A. Arbib [1975], "Eye-Movements and Visual Perception. A 'Two-Visual System' Model", Int. J. Man-Machine Studies, 7, 547-569.

L. Erman and V. Lesser [1975], "A Multi-Level Organization for Problem-Solving Using Many, Diverse, Cooperating Sources of Knowledge", Proc. 4th Int. Joint Conf. Artificial Intelligence, 483-490.

A. Hanson and E. Riseman [1974], "Preprocessing Cones: A Computational Structure for Scene Analysis", COINS Technical Report 74C-7, Department of Computer and Information Science, University of Massachusetts, Amherst.





A. Hanson and E. Riseman [1975], "The Design of a Semantically Directed Vision Processor (Revised and Updated)", COINS Technical Report 75C-1, Department of Computer and Information Science, University of Massachusetts, Amherst.

D. H. Hubel and T. N. Wiesel [1962], "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex", J. Neurophysiol., 28, 229-289.

B. Julesz [1971], Foundations of Cyclopean Perception, Chicago University Press.

M. D. Kelly [1971], "Edge Detection in Pictures by Computer Using Planning", Machine Intelligence 6, 379-409.

W. L. Kilmer, W. S. McCulloch and J. Blum [1969], "A Model of the Vertebrate Central Command System", Int. J. Man-Machine Studies, 1, 279-309.

A. Klinger and C. R. Dyer [1974], "Experiments on Picture Representation Using Regular Decomposition", Technical Report UCLA-ENG 7494, University of California, Los Angeles.

J.Y. Lettvin, H. Maturana, W. S. McCulloch and W. H. Pitts [1959], "What the Frog's Eye Tells the Frog's Brain", Proc. IRE, 47, 1940-1951.

D. Marr [1975], "Early Processing of Visual Information", AI Memo 340, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge.

W. S. McCulloch [1949], "A Heterarchy of Values Determined by the Topology of Nervous Nets", Bull. Math. Biophys., 11, 89-93.

M. L. Minsky [1975], "A Framework for Representing Knowledge", in The Psychology of Computer Vision (P. H. Winston, ed.), McGraw-Hill, 211-277.

F. S. Montalvo [1975], "Consensus vs. Competition in Neural Networks", Int. J. Man-Machine Studies, 7, 333-346.

A. Newell and H.A. Simon [1972], Human Problem Solving, Prentice-Hall.

J. D. Pettigrew, T. Nikara and P. O. Bishop [1968], "Binocular Interaction on Single Units in Cat Striate Cortex", Experimental Brain Research, 6, 391-410.

E. M. Riseman and M. A. Arbib [1976], "Computational Techniques in Visual Systems. Part II: Segmenting Static Scenes", Proc. IEEE.

A. Rosenfeld, R. A. Hummel, S. W. Zucker [1976], "Scene Labelling by Relaxation Operations", IEEE Trans. Systems, Man and Cybernetics, SMC-6, 420-433.

J. Ross [1976], "The Resources of Binocular Perception", Scientific American, March, 234, 80-86.

S. L. Tanimoto and T. Pavlidis [1975], "A Hierarchical Data Structure for Picture Processing", Computer Graphics and Image Processing, June

J. M. Tenenbaum and H. G. Barrow [1976], "Experiments in Interpretation-Guided Segmentation", Technical Note 123, Artificial Intelligence Center, Stanford Research Institute.

L. Uhr [1972], "Layered 'Recognition Cone' Networks that Preprocess, Classify, and Describe", IEEE Trans. Computers, C-21, 758-768.

D. Waltz [1975], "Understanding Line Drawings of Scenes with Shadows", in The Psychology of Computer Vision (P. H. Winston, ed.), McGraw-Hill, 19-91.

Y. Yakimovsky and J. A. Feldman [1973], "A Semantics-Based Decision Theory Region Analyzer", Proc. of Third Joint Conference on Artificial Intelligence, 580-588.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

**REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: COINS TR 76-10
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): COMPUTATIONAL TECHNIQUES IN VISUAL SYSTEMS PART I. THE OVERALL DESIGN
5. TYPE OF REPORT & PERIOD COVERED: INTERIM
6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): Michael A. Arbib, Edward M. Riseman
8. CONTRACT OR GRANT NUMBER(s): NIH 5R01 NS09755-06 COM, ONR N00014-75-C-0459, NSF DCR75-16098

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Computer & Information Science, University of Massachusetts, Amherst, MA 01002
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:

11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Arlington, VA 22217
12. REPORT DATE: 7/76
13. NUMBER OF PAGES: 60

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report): Distribution of this document is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
visual perception, computer vision, brain theory, schemas, representation of knowledge, cooperative computation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):
Our overall goal is to define computational techniques to be used by a system in making a visual scan of a dynamic environment with which it is to interact. Here, we discuss both brain mechanisms in the visual system of animals and humans and computer techniques for the analysis of color photographs of natural scenes. We present schemas as a formalization of the system's 'knowledge units'. This notion is helpful for our work in both the BT (Brain Theory) and AI (Artificial Intelligence)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

approaches. We further present specific studies--from our own group and from elsewhere--of subsystems of both animal and computer visual systems. We shall examine the interaction of high-level processes with low-level systems, as part of a general emphasis on integrated system design. Part II (Riseman and Arbib [1976]) will focus on techniques for segmenting single static colored images.